EMBEDDED OPEN SOURCE SUMMIT

Crafting a Vision-Aided Software Stack for UAV

Jim Huang (National Cheng Kung University, Taiwan)
Sheng-Wen Cheng (National Taiwan University, Taiwan)
April 16, 2024

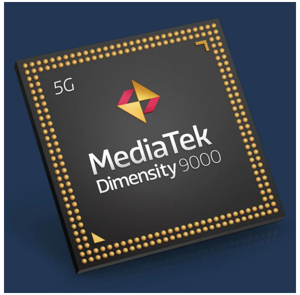
Motivations

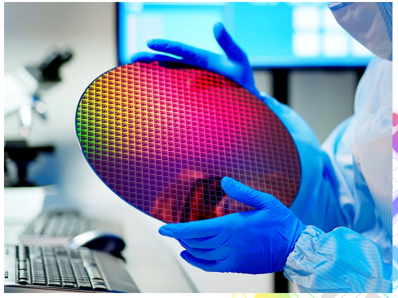
The most dangerous place on Earth

- As *The Economist* suggests that Taiwan is "the most dangerous place on Earth", Taiwanese has heavy needs on building our own national defense infrastructures, including UAVs.
- The Russo-Ukrainian War has shown to the world that the UAV technology has changed the form of modern warfare.
- Due to national defense considerations, Taiwanese government encourages corporations to innovate and develop UAV systems.

Strength of Taiwan in Hardware and IC design

- With more than 30 years of development, Taiwan has one of the mature IC/hardware supply chain in the world, including hardware design like microchips (MediaTek), power systems (Delta Electronics), industrial computer (Advantech), and IC manufacturing (TSMC).
- Taiwan has advantage to provide high-quality but low-cost hardware, which has potential in the UAV market for **mass production**.





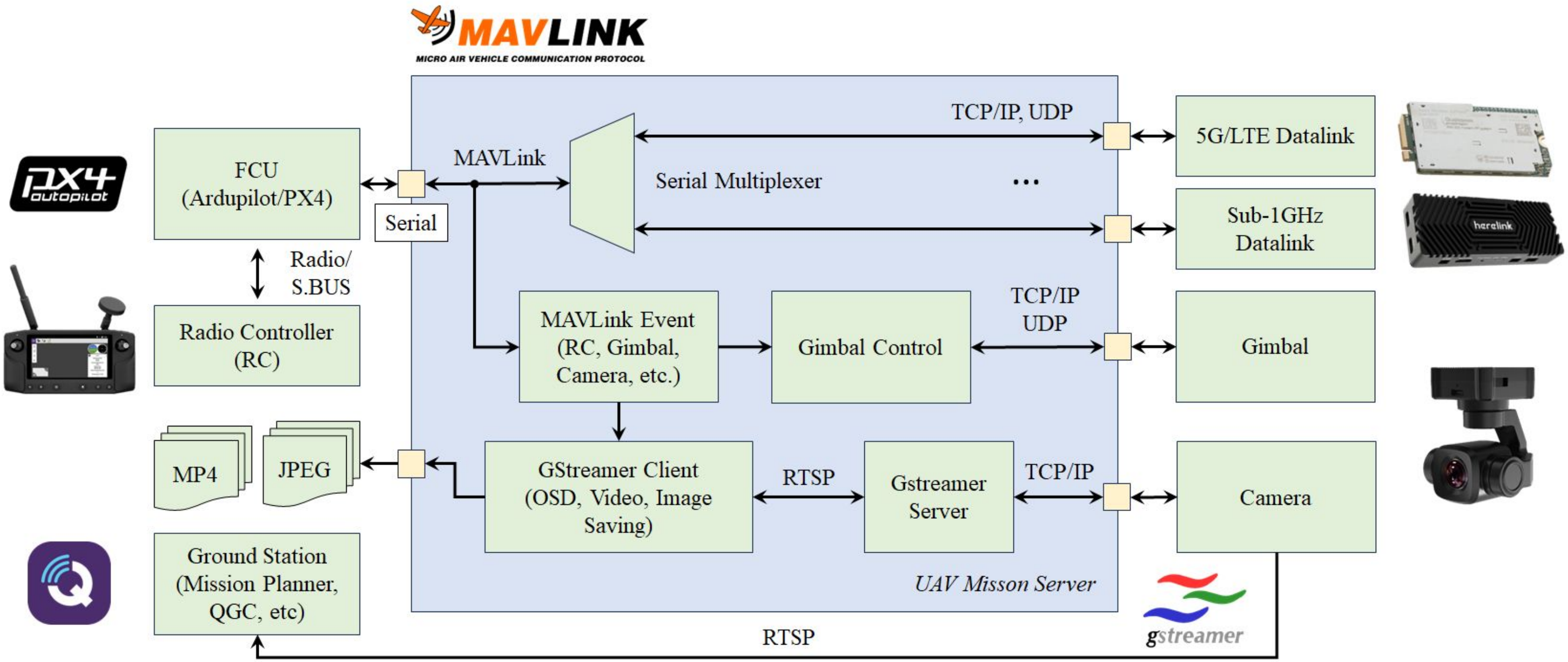
Issues that Taiwanese UAV manufacturers encountered

Through contacts with some Taiwanese UAV manufacturers from the university, we found that they have difficulties adopting open-source solutions in their products due to:

- Software-hardware integration, especially camera and gimbal systems (Requires certain level of code modification)
- Redundant data-link design with 5G LTE and Sub-1GHz modules
- Deploying high-level software stacks on low-cost computational hardware
- Middleware selections, the right scenario to use ROS, LCM, or MAVLink

We want to share our experience of resolving these problems and showcase a proof-of-concept design to the community.

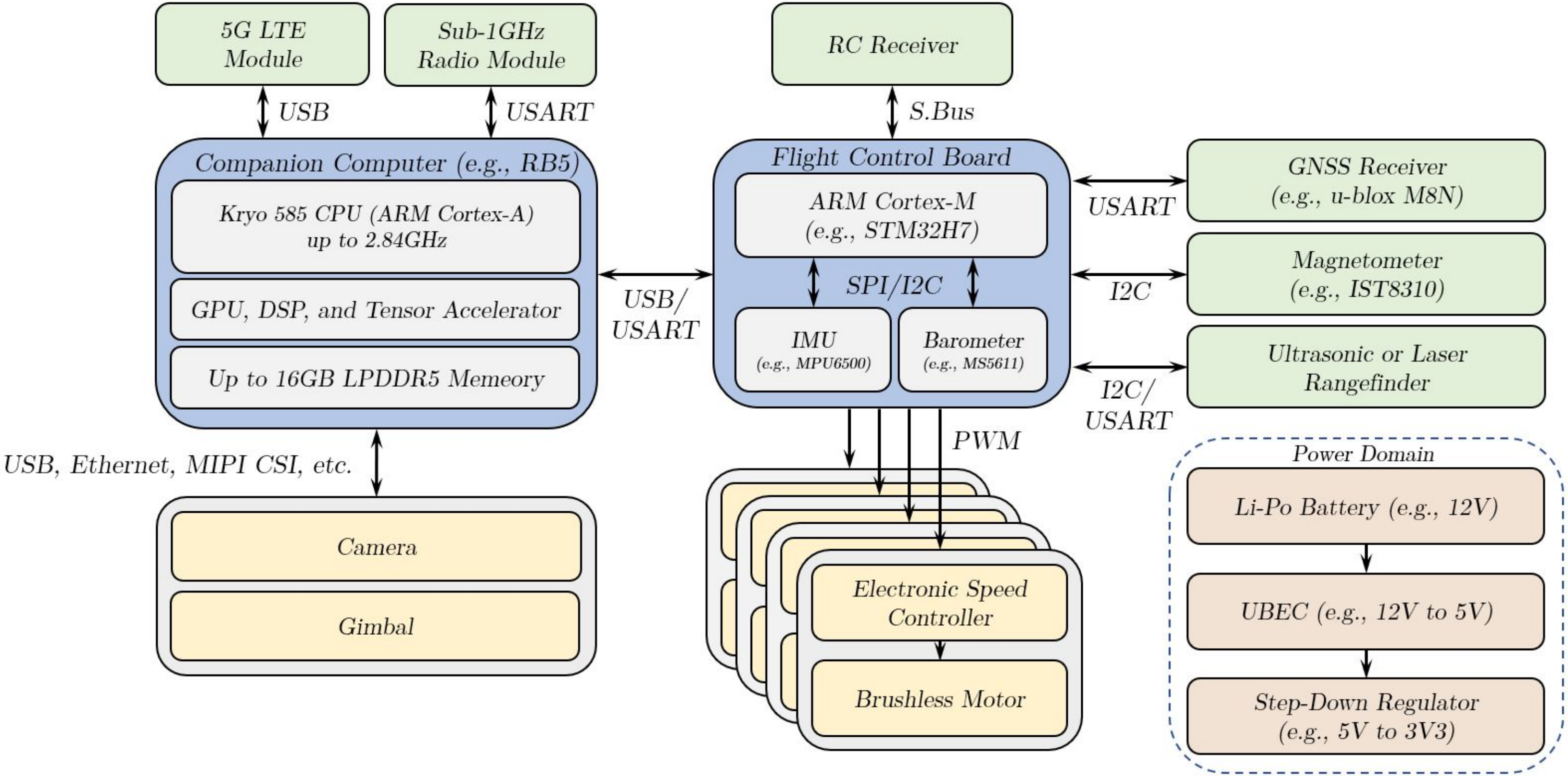
The proposed software: UAV Mission Server



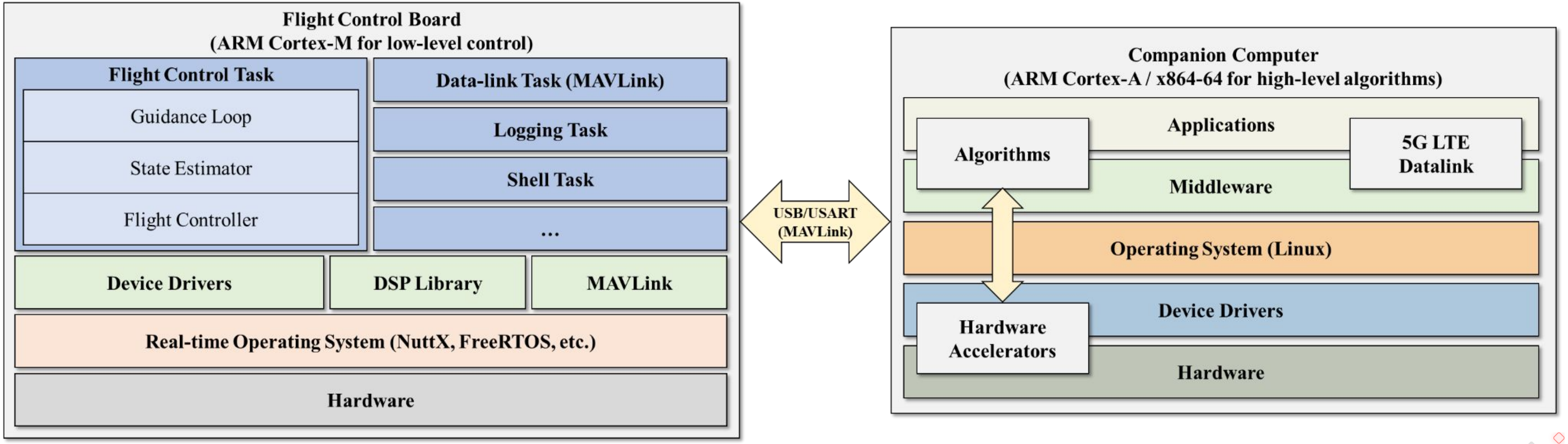
Overview of UAV system

Common building blocks of an UAV

UAV hardware architecture



Embedded Linux on UAVs

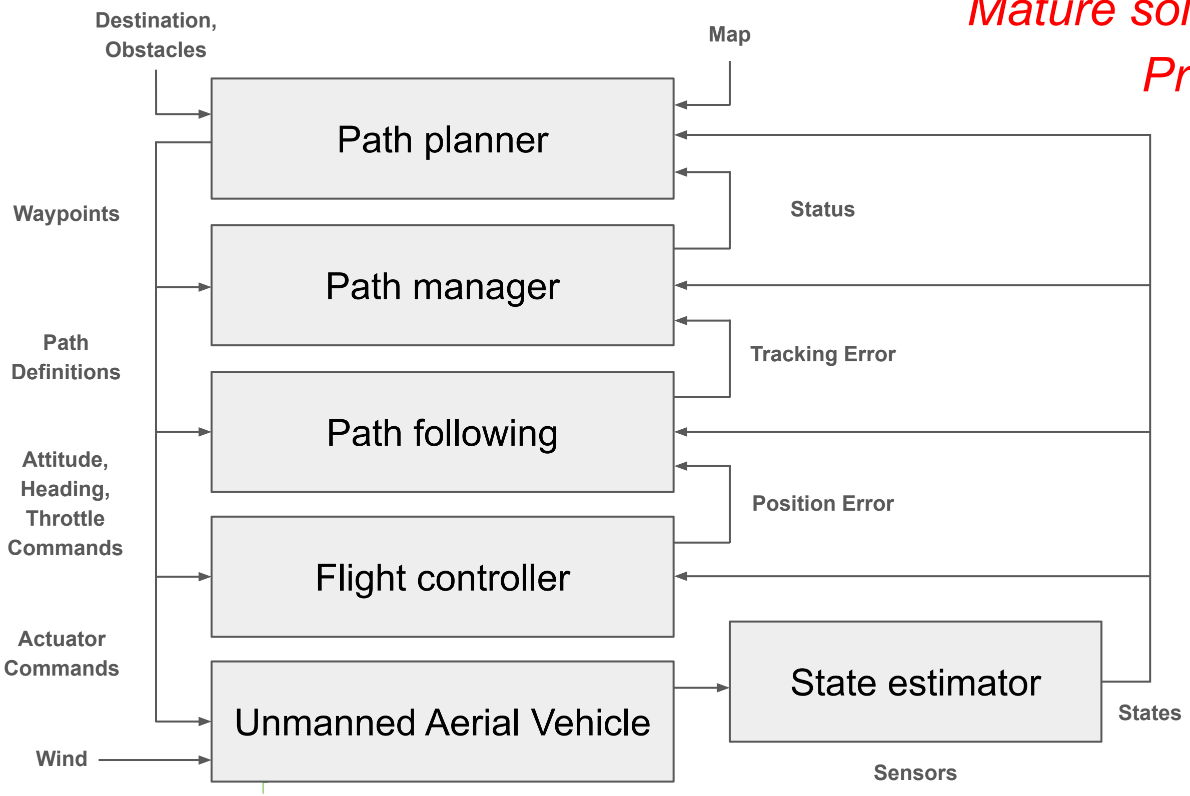


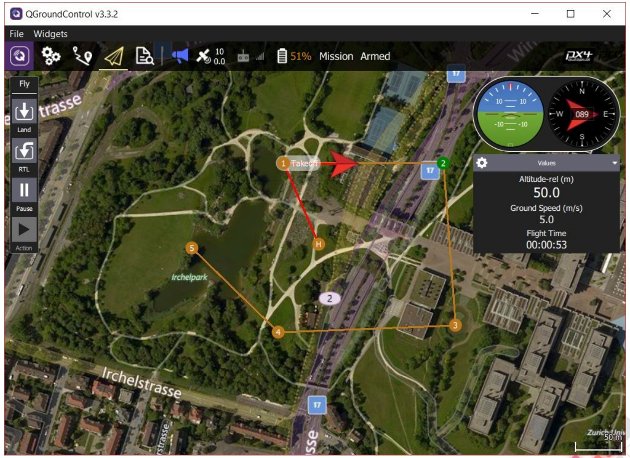
Opportunities for embedded Linux on UAVs

- Linux is usually running on a companion computer along with a flight control computer
- Flight control computer (e.g., PX4) usually has limited computing power and only take care of low level control tasks
- What people actually want is to let UAV performing high level tasks, these tasks are usually complicated and can only be run on Linux
- So far no unified project that combined all high-level needs of UAV into a single project (camera/gimbal control, streaming, deep learning, localization, mapping, etc.), which requires heavy works on integrating all separated features together.

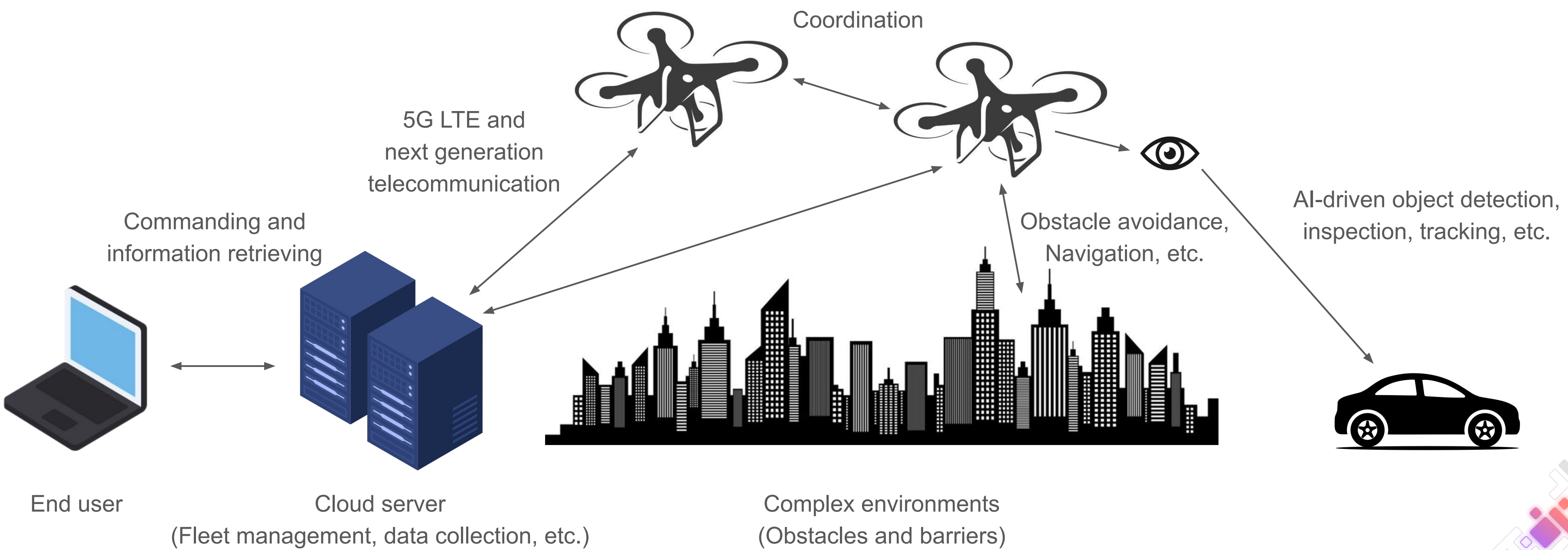
Traditional software design of UAV

Mature solution for deterministic plan / Pre-planned missions





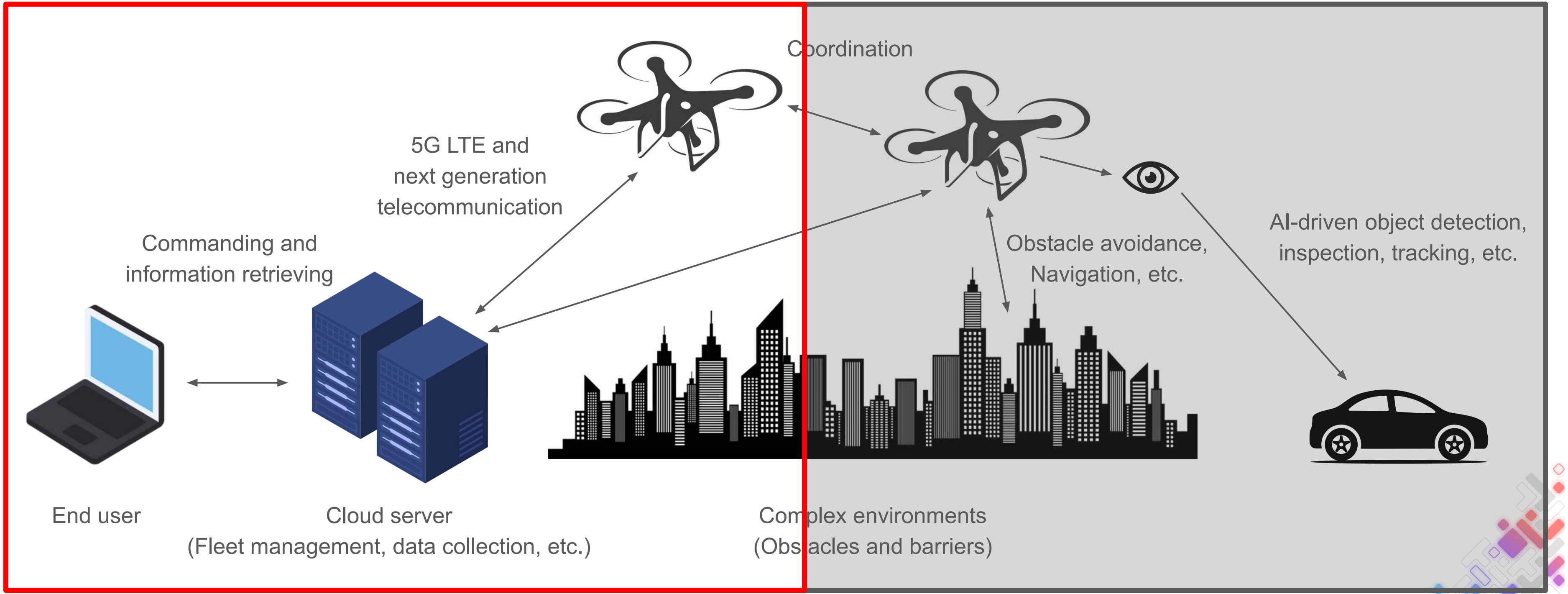
The Prospective Future of the evolving UAVs



Applications:

Inspection, tracking, transportation, medical, etc.

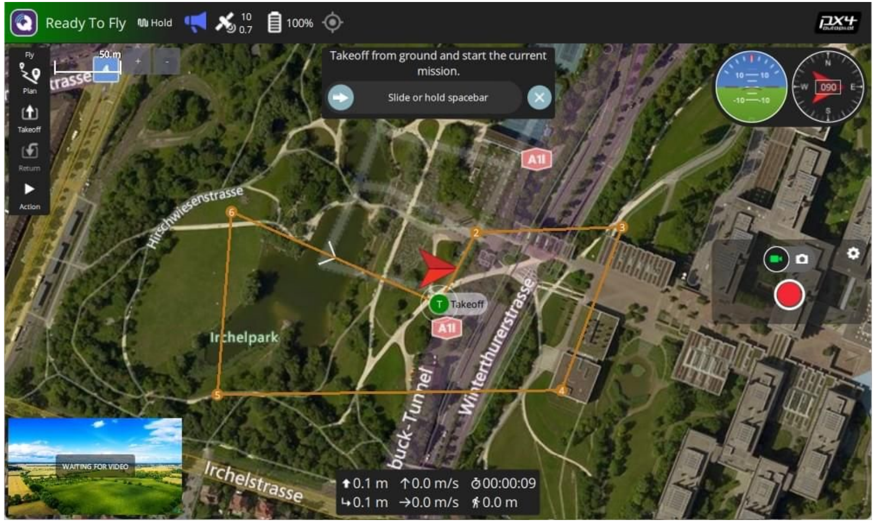
Data-link over air



Commanding and Messaging via MAVLink

Open-source ground stations

QGroundControl



MissionPlanner

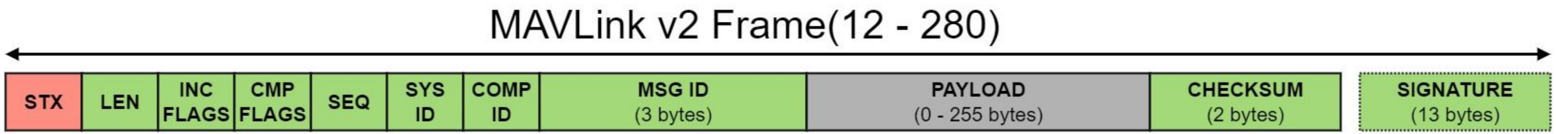
UAV ⟷ Ground Station

MAVLink

EOSS EMBEDDED OPEN SOURCE SUMMIT

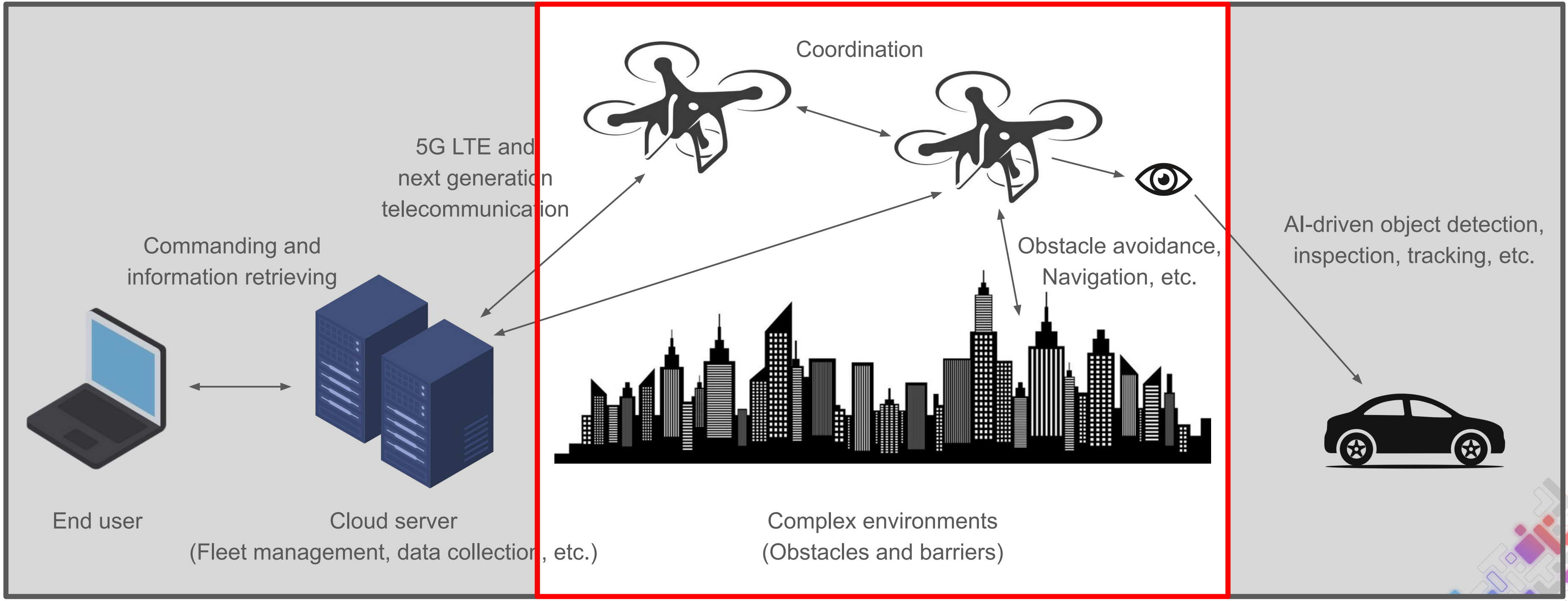
MAVLink

- A very lightweight messaging protocol for communicating with drones
- Suitable for UAV Real-time control
- Decentralized design as it follows publish-subscribe model and point-to-point design
- No restricted Transport Layer, can be sent via UART, UDP, TCP, CAN, etc.
- Widely adopted by the UAV community with great compatibility
- Support message checksum and signing for robustness and authentication



Data-link between UAV and ground station

Navigation, perception and planning in complex environment



High-level algorithms are typically modularized into different processes and inter-communicating via middleware like ROS or LCM.

SLAM (Simultaneous Localization and Mapping)

SLAM is a fundamental technology providing robot systems its current location and surrounding map

SLAM is usually based on LiDAR, Camera, IMU (Inertial Measurement Unit), or even more sensors.

Why using SLAM rather than just GPS?

- GPS signals (from satellite) are unreliable indoors when under obstruction.

Many advanced algorithms (e.g., object avoidance) require SLAM providing location and map information

LiDAR

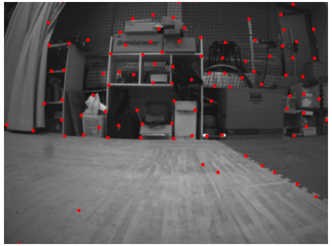
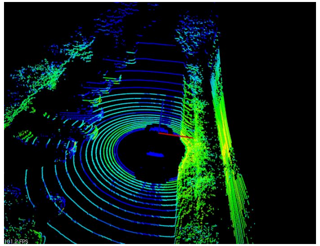
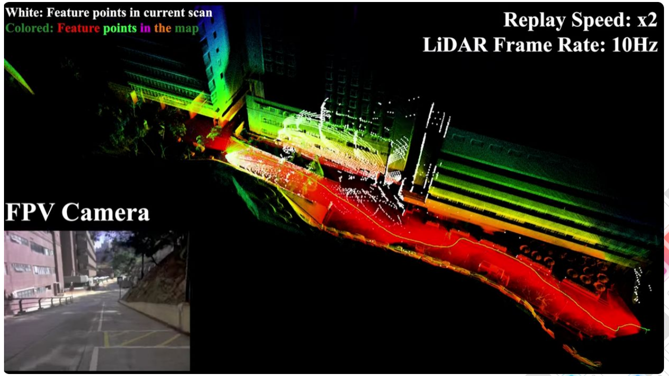


Image feature points

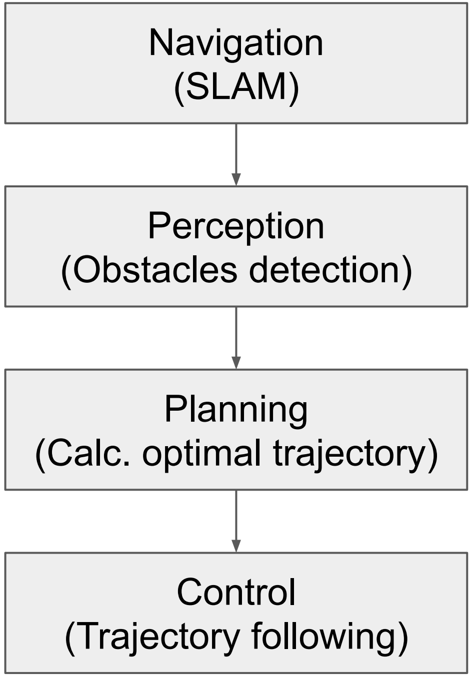


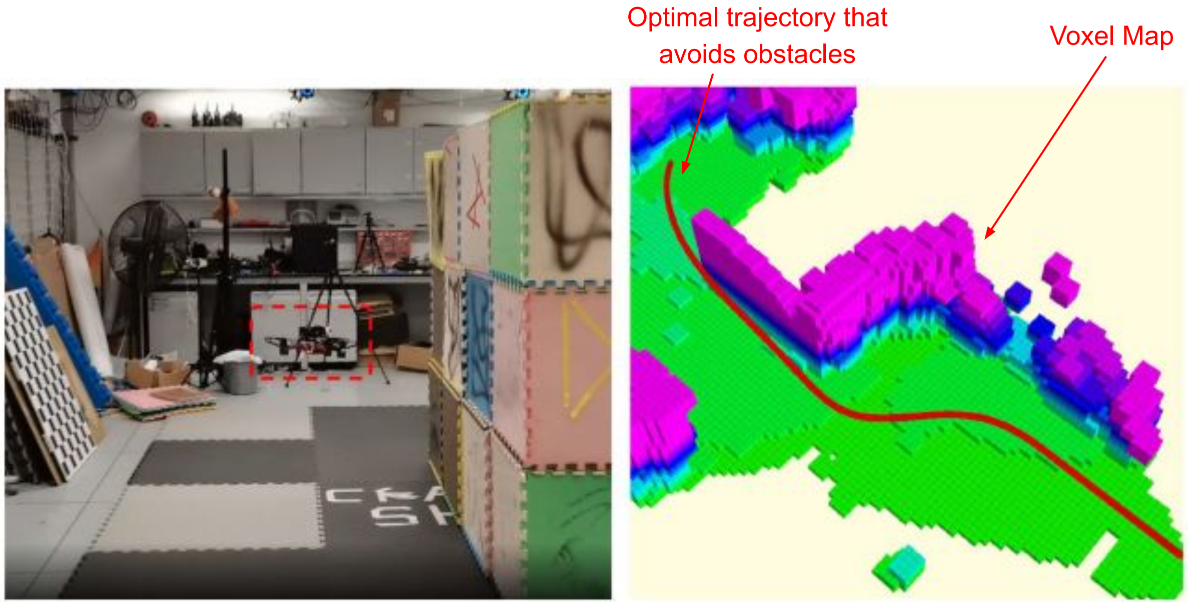
LiDAR point cloud



FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter

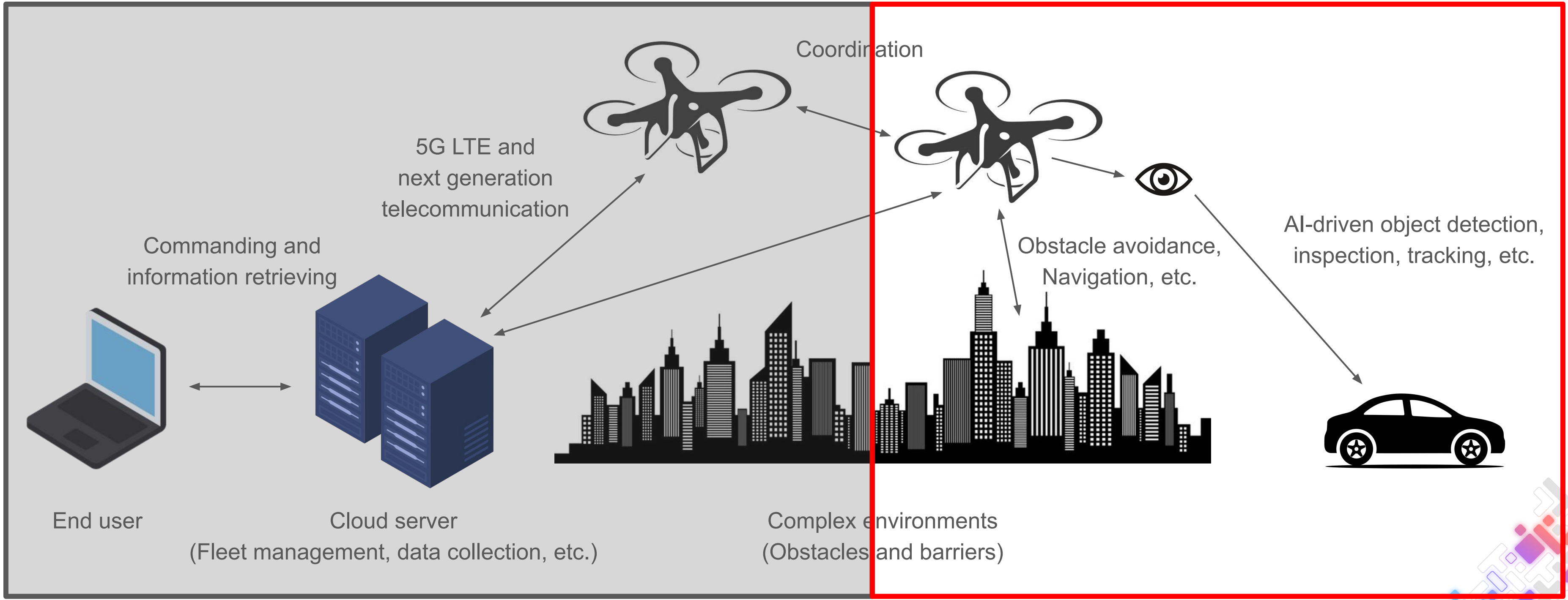
High-level task consisted of multiple algorithms





RAPTOR: Robust and Perception-Aware Trajectory Replanning for Quadrotor Fast Flight

Smart drone: AI-driven functionalities



Deep Learning on embedded platform is no longer formidable

Growing trend of AI in UAV applications

- Many useful applications: Surveillance, patrol, monitoring, rescue, etc.
- Some of the current embedded platforms can run small DL models (visual application) within 10~30 Hz.
- AI acceleration (e.g., Qualcomm RB5/RB6, Nvidia TensorRT).

UAV-Based Crop and Weed Classification for Smart Farming

Video: CM62 High Speed Object Tracking (AVT Australia)

Open-source UAV autopilots

EMBEDDED
OPEN SOURCE
SUMMIT

Open-source autopilots of UAVs

ARDUPILOT

AUTO QUAD
AUTONOMOUS MULTI ROTOR VEHICLE CONTROLLER

Paparazzi

PX4 autopilot

BETAFLIGHT

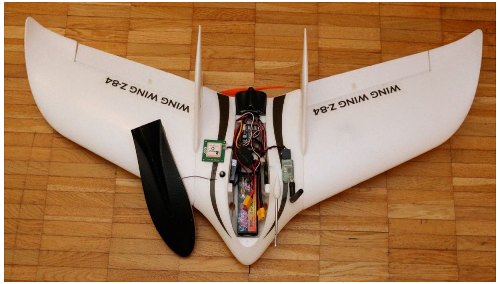
bitcraze

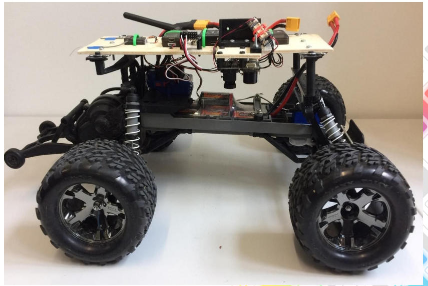
PX4

Dronecode FOUNDATION

PX4 autopilot

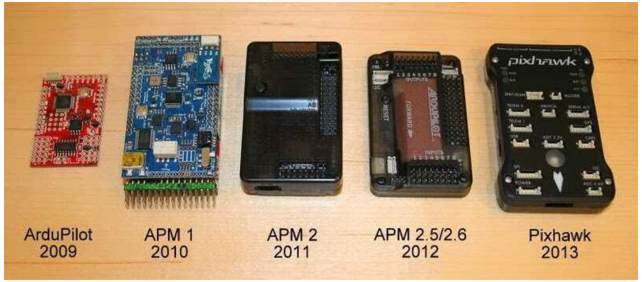
- Originating in 2008 by Lorenz Meier
- PX4 refers to the autopilot software where Pixhawk is the hardware name
- Hosted by Dronecode Project to collaborate with Linux foundation since 2014
- Rich vehicle types support: Multirotor, Fixed Wing, VTOL, Helicopter, Rover, etc.
- Written in C++ and based on NuttX
- Released under BSD license





ArduPilot

- Released in 2009 by Jordi Muñoz (with Chris Anderson they founded the 3D Robotics)
- Originally based on Arduino and later migrated to other platforms
- Written in C and based on ChibiOS (a RTOS with Hardware Abstraction Layer)
- Compatible with many Pixhawk hardware
- Released under GPLv3 license



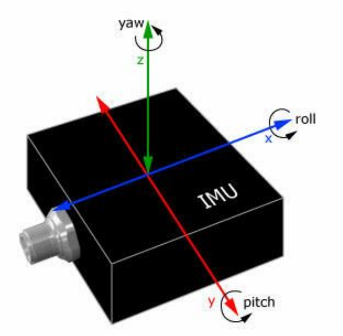
3DR Solo was a commercial product launched in 2015, which was based on ArduCopter 3.3 software and Pixhawk 2.0 hardware.

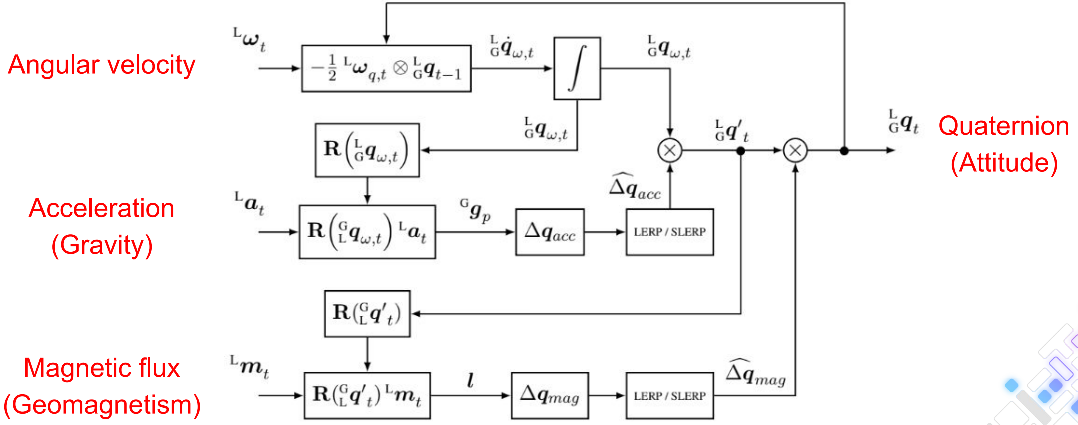
Autopilot internal: State estimator

Goal: Obtain information like ***position***, ***velocity***, ***attitude***, ***sensor bias***, etc.

Sensor: Accelerometer, Gyroscope, GPS receiver, Magnetometer, etc.

Algorithms: Kalman Filter, Complementary Filter, Gradient-Descent Method, etc.



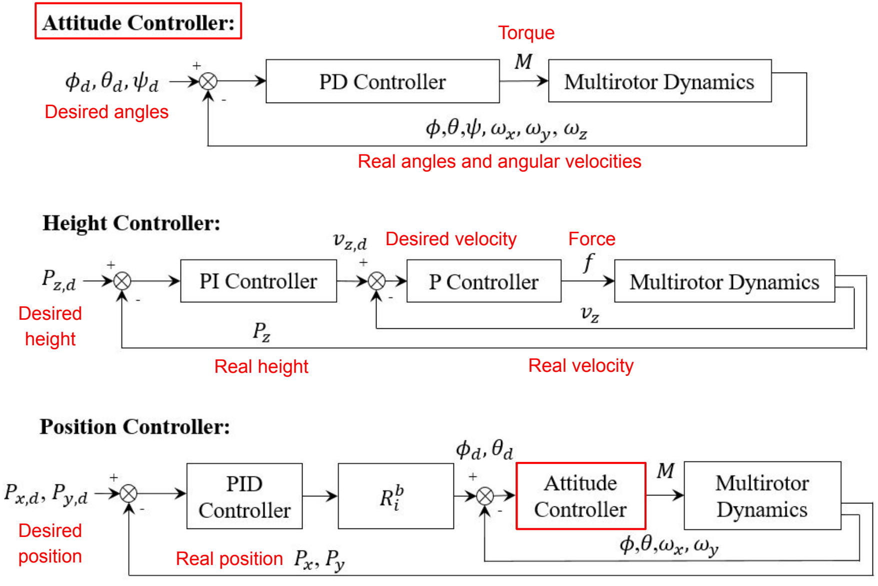


Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs

Autopilot internal: Feedback controller

Stabilizing control variables to their desired setpoints via a feedback loop design

Control variables of multirotor UAV: position, velocity, attitude, angular rate



* R_i^b: Rotational transform from earth frame to body-fixed frame

Quadrotor controllers are usually designed in a decoupled fashion with a rotational part and a translational part.

Intuitively, a Quadrotor can only move in space by changing its own attitude.

Choosing a proper Middleware

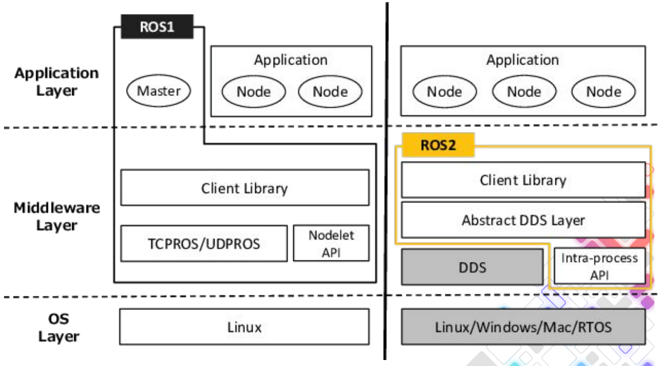
ROS (Robot Operating System)

ROS1 (Deprecated in 2025)

- Based on TCPROS/UDPROS protocol for communication between nodes
- Master-Client architecture
- Limited real-time capabilities, not suitable for hard real-time applications
- Single thread design of ros::spin()

ROS2

- DDS (Data Distribution Service) based communication
- Improved real-time capabilities
- Offers Quality of Service (QoS) for adjusting reliability, durability, and deadline for different services
- Support multi-threaded executor



Exploring the Performance of ROS2

LCM (Lightweight Communications and Marshalling)

Initially designed for MIT's research of autonomous vehicle in DARPA Urban Challenge.

- Low-latency inter-process communication
- Efficient broadcast mechanism using UDP Multicast
- Supports multiple programming language just like ROS

Some features that may make it be favorable than ROS:

- Requires only few dependencies
- Platform: Linux, macOS, Windows, **POSIX-1.2001 OS** (e.g., Cygwin, BSD, etc.)

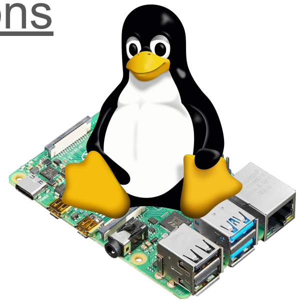
A software stack that just fit the needs

ROS has rich features and a large software ecosystem behinds it.

However, installing ROS on non-Ubuntu Linux distribution is not an easy job.

- Not all SoC vendors provide Ubuntu as it requires license fee for Canonical (Especially for those cheap boards you can find)
- Size considerations: Ubuntu and ROS are relatively large. For many case, designers only want a minimal system like using Buildroot to just fit the needs

LCM maybe a good replacement under such considerations



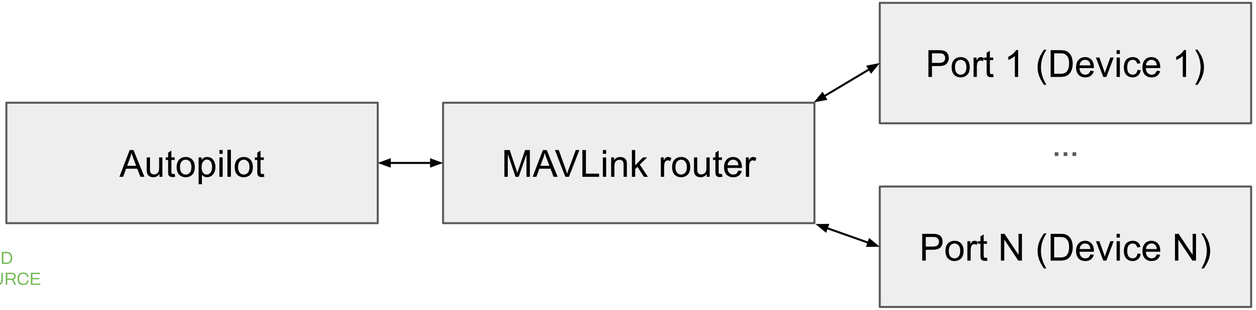
System integration issues

Camera and gimbal

- Various type of interfaces exist cause the inconvenient of integration:
  - Camera interface: USB, Ethernet, MIPI CSI, etc.
  - Gimbal interface: USB, UART, etc.
- Some open standard like STorM32 (a open source gimbal design) does exist but are not an unified solution
- In reality, the camera and gimbal may be decoupled or coupled as a single product depending on the design.
- Even if the camera is capable of streaming, it may not utilize the hardware acceleration feature on the SoC
- The developer may need to write some simple drivers before developing applications like computer vision or deep learning.

Redundant data-link channels via MAVLink router

- As a safety-critical system, robust communication is essential for UAVs
- One common approach is to offer extra backup channel when the major data-link source breaks down (e.g., 5G LTE + Sub-1GHz radio)
- A MAVLink router may be used to solve such the problem. The router receives byte-data from multiple ports and only forwards if an entire message is decoded successfully



Proposed system:
A reference design

UAV Mission Server

github.com/shengwen-tw/uav-mission-server

Running on the companion computer

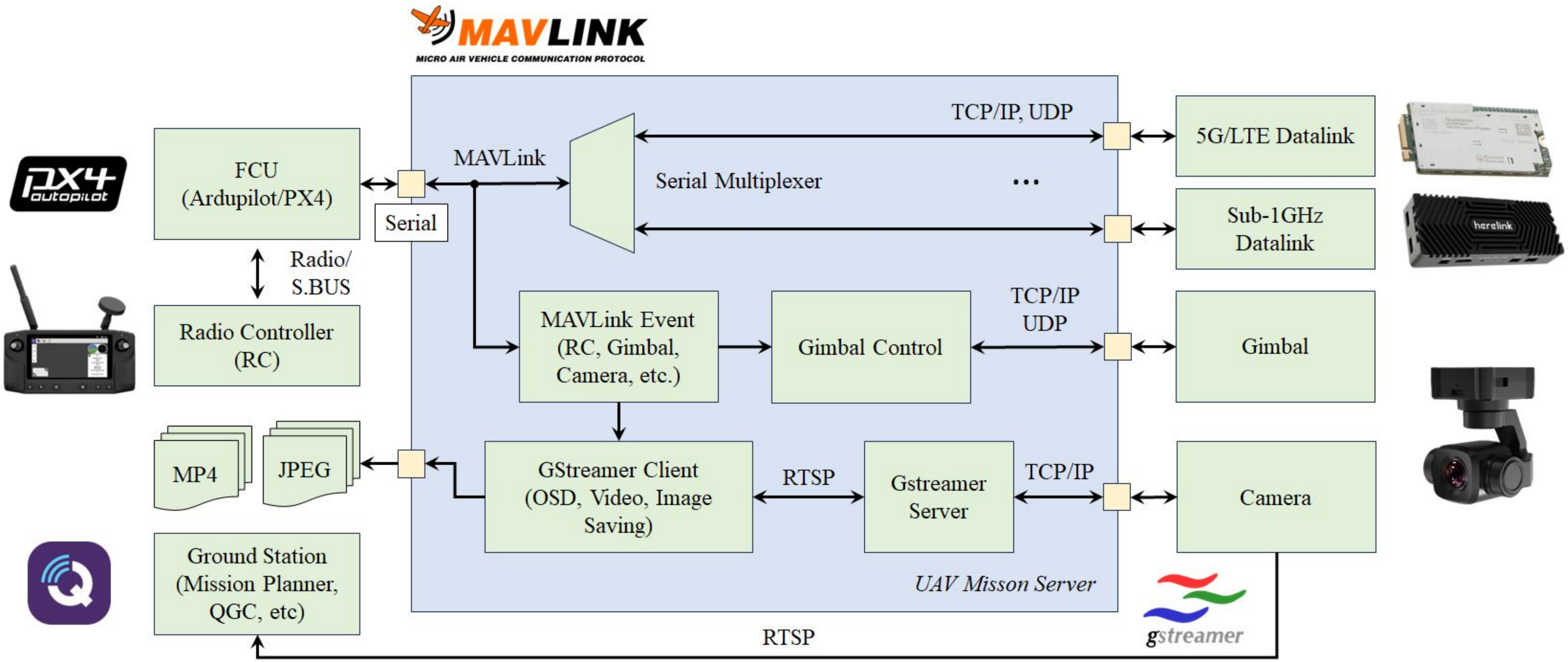
So far a proof-of-concept project with the following features:

- RTSP video streaming via GStreamer (H.264, H.265, V4L2, etc.)
- Integrated with MAVLink to interact with ground stations (image/video capturing, gimbal control, etc.)
- A Hardware Abstraction Layer for camera and gimbal devices
- Allow redundant data-link channels via concept of MAVLink router (Currently 5G LTE and sub-1GHz telemetry)

Similar projects:

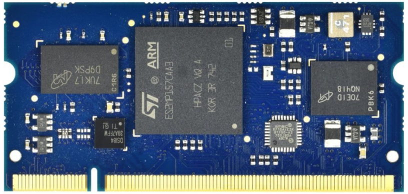
- mavlink-camera-manager (written in Rust!)
- Dronecode Camera Manager (deprecated)

Software Architecture of the UAV Mission Server

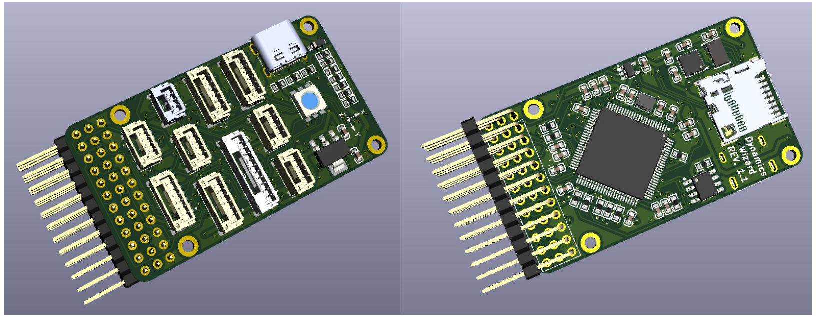


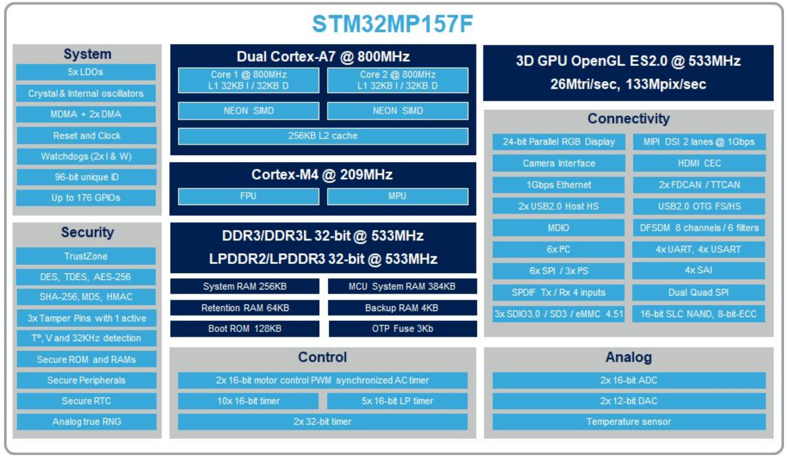
Related and future work

Multi-core integration on single SoC



- Some SoC vendors start offering single SoC with ARM Cortex-A + Cortex-M
- SoCs like STM32MP15 product line offering (A7@800MHz + M4@209MHz)
- Appealing to entry-level products with basic internet connectivity, video streaming, etc.
- We are working on a lightweight and highly-customized flight control software to build such a system, and may release the source code in the future.





EOSS

EMBEDDED
OPEN SOURCE
SUMMIT